AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A computer-implemented user interface configuration method, comprising:

detecting a user proficiency level with respect to a user interface, based on user behavior with respect to the user interface; and

automatically configuring at least one functional component of the user interface responsive to the detected proficiency level.

2. (Currently amended) The method of claim 1, wherein automatically configuring the at least one functional component of the user interface comprises:

selecting at least one configuration option from a plurality of configuration options.

3. (Currently amended) The method of claim 1, wherein automatically configuring the at least one functional component of the user interface comprises at least one selected from the group consisting of:

enabling access to a functional user interface element;

disabling access to a functional user interface element; and

changing an appearance of a functional user interface element.

4. (Currently amended) The method of claim 1, wherein automatically configuring the at least one functional component of the user interface comprises at least one selected from the group consisting of:

enabling access to a command;

disabling access to a command;

changing an appearance of a command;

enabling access to a menu;

disabling access to a menu;

changing an appearance of a menu;

enabling access to a button;

disabling access to a button;

changing an appearance of a button;

enabling access to a shortcut; and

disabling access to a shortcut~~; and~~

~~changing an appearance of a command~~.

5. (Cancelled).

6. (Cancelled).

7. (Cancelled).

8. (Cancelled).

9. (Cancelled).

10. (Cancelled).

11. (Original) The method of claim 1, further comprising:

outputting a notification of a change to user interface configuration.

12. (Original) The method of claim 1, further comprising:

outputting a notification of at least one newly enabled user interface feature.

13. (Original) The method of claim 1, wherein detecting the user proficiency level and automatically configuring the user interface are performed responsive to a trigger event.

14. (Original) The method of claim 13, wherein the trigger event comprises user input requesting user interface configuration.

15. (Original) The method of claim 13, wherein the trigger event comprises application startup.

16. (Original) The method of claim 13, wherein the trigger event comprises system startup.

17. (Original) The method of claim 13, wherein the trigger event comprises a change in user behavior with respect to the user interface.

18. (Original) The method of claim 13, wherein the trigger event comprises user logon.

19. (Currently amended) The method of claim 1, wherein detecting the user proficiency level and automatically configuring the at least one functional component of the user interface are performed periodically.

20. (Original) The method of claim 1, wherein detecting the user proficiency level comprises reading a stored user proficiency level derived from at least one marker.

21. (Original) The method of claim 20, wherein the marker indicates historical usage of the user interface.

22. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting whether a user interface element has been used.

23. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting whether a user interface element has been used a number of times exceeding a predetermined threshold.

24. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting a total amount of time spent by a user using an application.

25. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting how many applications are open concurrently.

26. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting a historical average number of concurrently open applications.

27. (Cancelled).

28. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting how many windows are open concurrently.

29. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting a historical average number of concurrently open windows.

30. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting a user-specified preference indicating a proficiency level.

31. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting web page visitation patterns.

32. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting historical usage of secure web pages.

33. (Original) The method of claim 1, wherein detecting the user proficiency level comprises detecting historical usage of web pages having active content.

34. (Currently amended) The method of claim 1, wherein:

detecting the user proficiency level comprises detecting the user proficiency level with respect to a user interface component less than the entire user interface; and

automatically configuring the <u>at least one functional component of the</u> user interface comprises automatically configuring the user interface component without altering the configuration of the remainder of the user interface.

35. (Currently amended) The method of claim 1, wherein:

detecting the user proficiency level comprises detecting the user proficiency level with respect to an application; and

automatically configuring the at least one functional component of the user interface comprises automatically configuring the user interface for the application.

36. (Original) The method of claim 1, further comprising:

responsive to user behavior with respect to the user interface, storing a marker indicating a user proficiency level;

and wherein detecting the user proficiency level comprises reading the stored marker.

37. (Original) The method of claim 36, wherein:

storing the marker is performed by a first application; and

reading the stored marker is performed by a background process.

38. (Original) The method of claim 36, wherein:

storing the marker is performed by a first application; and

reading the stored marker is performed by a second application different from the first application.

39. (Original) The method of claim 36, wherein:

storing the marker is performed by an operating system; and

reading the stored marker is performed by the operating system.

40. (Currently amended) The method of claim 39, wherein:

automatically configuring the at least one functional component of the user interface comprises modifying functional user interface elements that are supplied to a plurality of applications.

41. (Original) The method of claim 36, wherein:

storing the marker is performed by an operating system; and

reading the stored marker is performed by an application.

42. (Original) The method of claim 1, wherein detecting the user proficiency level comprises retrieving a plurality of stored markers and aggregating the retrieved markers to derive a proficiency level.

43. (Original) The method of claim 1, further comprising:

responsive to user behavior with respect to the user interface, storing a plurality of markers;

and wherein detecting the user proficiency level comprises retrieving at least a subset of the stored markers and aggregating the retrieved markers to derive a proficiency level.

44. (Original) The method of claim 1, further comprising:

accepting user input overriding the user interface configuration and specifying a desired configuration; and

responsive to the user input, configuring the user interface according to the desired configuration.

45. (Currently amended) The method of claim 1, wherein:

detecting a user proficiency level with respect to a user interface comprises detecting a user proficiency level with respect to a user interface of a web-resident application being run from a client machine; and

automatically configuring the at least one functional component of the user interface comprises automatically configuring at least one functional user interface element for the web-resident application.

46. (Currently amended) A computer program product for configuring a user interface, comprising:

a computer-readable medium; and

computer program code, encoded on the medium, for:

detecting a user proficiency level with respect to a user interface, based on user behavior with respect to the user interface; and

automatically configuring at least one functional component of the user interface responsive to the detected proficiency level.

47. (Currently amended) The computer program product of claim 46, wherein the computer program code for automatically configuring the at least one functional component of the user interface comprises computer program code for:

selecting at least one configuration option from a plurality of configuration options.

48. (Currently amended) The computer program product of claim 46, wherein the computer program code for automatically configuring the at least one functional component of the user interface comprises at least one selected from the group consisting of:

computer program code for enabling access to a functional user interface element;

computer program code for disabling access to a functional user interface element; and

computer program code for changing an appearance of a functional user interface element.

49. (Currently amended) The computer program product of claim 46, wherein the computer program code for automatically configuring the at least one functional component of the user interface comprises at least one selected from the group consisting of:

computer program code for enabling access to a command;

computer program code for disabling access to a command;

computer program code for changing an appearance of a command;

computer program code for enabling access to a menu;

computer program code for disabling access to a menu;

computer program code for changing an appearance of a menu;

computer program code for enabling access to a button;

computer program code for disabling access to a button;

computer program code for changing an appearance of a button;

computer program code for enabling access to a shortcut; and

computer program code for disabling access to a shortcut~~; and computer program code for changing an appearance of a command~~.

50. (Cancelled).

51. (Currently amended) The computer program product of claim 46, wherein the computer program code for detecting the user proficiency level and automatically configuring the at least one functional component of the user interface comprises computer program code for performing the detecting and configuring steps responsive to a trigger event.

52. (Currently amended) The computer program product of claim 46, wherein the computer program code for detecting the user proficiency level and automatically configuring the at least one functional component of the user interface comprises computer program code for performing the detecting and configuring steps periodically.

53. (Original) The computer program product of claim 46, wherein the computer program code for detecting the user proficiency level comprises computer program code for reading a stored user proficiency level derived from at least one marker.

54. (Currently amended) The computer program product of claim 46, wherein:

the computer program code for detecting the user proficiency level comprises computer program code for detecting the user

proficiency level with respect to a user interface component less than the entire user interface; and

the computer program code for automatically configuring the at least one functional component of the user interface comprises computer program code for automatically configuring the functional user interface component without altering the configuration of the remainder of the user interface.

55. (Currently amended) The computer program product of claim 46, wherein:

the computer program code for detecting the user proficiency level comprises computer program code for detecting the user proficiency level with respect to an application; and

the computer program code for automatically configuring the at least one functional component of the user interface comprises computer program code for automatically configuring the user interface for the application.

56. (Original) The computer program product of claim 46, further comprising:

computer program code for, responsive to user behavior with respect to the user interface, storing a marker indicating a user proficiency level;

and wherein the computer program code for detecting the user proficiency level comprises computer program code for reading the stored marker.

57. (Original) The computer program product of claim 46, wherein the computer program code for detecting the user proficiency level comprises computer program code for retrieving a plurality of stored markers and aggregating the retrieved markers to derive a proficiency level.

58. (Original) The computer program product of claim 46, further comprising:

computer program code for, responsive to user behavior with respect to the user interface, storing a plurality of markers;

and wherein the computer program code for detecting the user proficiency level comprises computer program code for retrieving at least a subset of the stored markers and aggregating the retrieved markers to derive a proficiency level.

59. (Currently amended) The computer program product of claim 46, wherein:

the computer program code for detecting a user proficiency level with respect to a user interface comprises computer program code for detecting a user proficiency level with respect to a

user interface of a web-resident application being run from a client machine; and

the computer program code for automatically configuring the at least one functional component of the user interface comprises computer program code for automatically configuring at least one functional user interface element for the web-resident application.

60. (Currently amended) A system for configuring a user interface, comprising:

means for detecting a user proficiency level with respect to a user interface, based on user behavior with respect to the user interface; and

means for automatically configuring at least one functional component of the user interface responsive to the detected proficiency level.

61. (Currently amended) A system for configuring a user interface, comprising:

a user proficiency level detector, for detecting a user proficiency level with respect to a user interface, based on user behavior with respect to the user interface; and

a user interface configuration module, coupled to the user proficiency level detector, for automatically configuring at least one functional component of the user interface responsive to the detected proficiency level.

62. (Original) The system of claim 61, wherein the user interface configuration module selects at least one configuration option from a plurality of configuration options.

63. (Currently amended) The system of claim 61, wherein the user interface configuration module performs at least one selected from the group consisting of:

enabling access to a functional user interface element;

disabling access to a functional user interface element; and

changing an appearance of a functional user interface element.

64. (Currently amended) The system of claim 61, wherein the user interface configuration module performs at least one selected from the group consisting of:

enabling access to a command;

disabling access to a command;

changing an appearance of a command;

enabling access to a menu;

disabling access to a menu;

changing an appearance of a menu;

enabling access to a button;

disabling access to a button;

changing an appearance of a button;

enabling access to a shortcut; and

disabling access to a shortcut~~; and~~

~~changing an appearance of a command~~.

65. (Cancelled).

66. (Original) The system of claim 61, wherein the user proficiency level detector and the user interface configuration module operate responsive to a trigger event.

67. (Original) The system of claim 61, wherein the user proficiency level detector and the user interface configuration module operate periodically.

68. (Original) The system of claim 61, wherein the user proficiency level detector reads a stored user proficiency level derived from at least one marker.

69. (Currently amended) The system of claim 61, wherein:

the user proficiency level detector detects the user proficiency level with respect to a user interface component less than the entire user interface; and

the user interface configuration module automatically configures the at least one functional component of the user interface component without altering the configuration of the remainder of the user interface.

70. (Currently amended) The system of claim 61, wherein:

the user proficiency level detector detects the user proficiency level with respect to an application; and

the user interface configuration module automatically configures the at least one functional component of the user interface for the application.

71. (Original) The system of claim 61, further comprising:

a marker storage device, for, responsive to user behavior with respect to the user interface, storing a marker indicating a user proficiency level;

wherein the user proficiency level detector reads the stored marker from the marker storage device.

72. (Original) The system of claim 61, wherein the user proficiency level detector retrieves a plurality of stored markers and aggregates the retrieved markers to derive a proficiency level.

73. (Original) The system of claim 61, further comprising:

a marker storage device, for, responsive to user behavior with respect to the user interface, storing a plurality of markers;

wherein the user proficiency level detector retrieves at least a subset of the stored markers and aggregates the retrieved markers to derive a proficiency level.

74. (Currently amended) The system of claim 61, wherein:

the user proficiency level detector detects a user proficiency level with respect to a user interface of a web-resident application being run from a client machine; and

the user interface configuration module automatically configures at least one <u>functional</u> user interface element for the web-resident application.